

Structure and Interpretation of Computer Programs

In the card game known as “Blackjack” or “Twenty-one,” each card from a deck of playing cards is assigned a numerical score. Tens, jacks, queens, and kings are each worth 10 points; aces are worth either 1 or 11 at the player’s discretion; and the remaining cards, deuces through nines, have scores equal to their ranks. The score of a hand equals the sum of the scores of its cards.

Assume there are two players – *a player* and *a dealer*. The objective of the player is to beat the dealer by getting the score of the hand as close to 21 as possible, without going over 21. The rules of the game are as follows.

The player and the dealer receive two cards from a shuffled deck. After the first two cards are dealt to dealer and player, the player is asked if she’d like another card (called *'hitting'*), or if she is happy with the cards she has already (called *'staying'*). The object is to make the sum of the player’s card values as close to 21, without going over. If the player makes 21 exactly, she has *blackjack*, which marks the end of the game and the player wins. If the player goes over 21, she *'busts'* marking the end of the game and the dealer wins. The player can stop *hitting* at any point and *stay*. Once the player’s turn is over, the dealer tries to do the same. The dealer must keep hitting until she gets to 17. If she gets above 17, the dealer busts and the player win (a variant of real game BTW!!). The dealer also can choose to stay and pass the turn to the player. The game continues until either of them busts and the other wins.

Your task is to implement the above “Blackjack” or “Twenty-one,” game in racket. You are expected to design appropriate data abstraction for a card, the deck and the game itself. You are also expected to realize the game play through appropriate procedural abstraction.